In [1]:
```
import pandas as pd
```

In [2]:
```
# 2 main datatypes
series = pd.Series(['BMW', 'Toyota', 'Honda'])
```

In [3]:
```
series
```

Out[3]:
```
0       BMW
1    Toyota
2     Honda
dtype: object
```

In [4]:
```
# series = 1-dimensional
```

In [5]:
```
colours = pd.Series(['red', 'blue', 'white'])
colours
```

Out[5]:
```
0      red
1     blue
2    white
dtype: object
```

In [6]:
```
# Dataframe = 2-dimentional
car_data = pd.DataFrame({'car make': series, 'colours': colours})
car_data
```

Out[6]:

	car make	colours
0	BMW	red
1	Toyota	blue
2	Honda	white

In [7]:
```
# import data
car_sales = pd.read_csv('car-sales.csv')
car_sales
```

Out[7]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [8]:
```
# exporting a dataframe
car_data.to_csv('exported-car-sales.csv', index = False)
```

In [9]:
```
exported_car_sales = pd.read_csv('exported-car-sales.csv')
exported_car_sales
```

Out[9]:

	car make	colours
0	BMW	red
1	Toyota	blue
2	Honda	white

In [10]:
```
# 2 exporting a dataframe
car_sales.to_csv('exported-2nd-car-sales.csv', index=False)
```

In [11]:
```
exported_2nd_car_sales = pd.read_csv('exported-2nd-car-sales.csv')
exported_2nd_car_sales
```

Out[11]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

Describe Data

In [12]:
```
# Attributes (it has no bracket)
car_sales.dtypes

# function
# car_sales.to_csv(): it has a bracket
```

Out[12]:
```
Make             object
Colour           object
Odometer (KM)     int64
Doors             int64
Price            object
dtype: object
```

In [13]:
```
1 car_sales.columns
```

Out[13]:
```
Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='object')
```

In [14]:
```
1 car_columns = car_sales.columns
2 car_columns
```

Out[14]:
```
Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='object')
```

In [15]:
```
1 car_sales.index
```

Out[15]:
```
RangeIndex(start=0, stop=10, step=1)
```

In [16]:
```
1 car_sales
```

Out[16]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [17]:
```
1 car_sales.describe
```

Out[17]:
```
<bound method NDFrame.describe of      Make Colour  Odometer (KM)  Doors        Price
0  Toyota  White         150043      4   $4,000.00
1   Honda    Red          87899      4   $5,000.00
2  Toyota   Blue          32549      3   $7,000.00
3     BMW  Black          11179      5  $22,000.00
4  Nissan  White         213095      4   $3,500.00
5  Toyota  Green          99213      4   $4,500.00
6   Honda   Blue          45698      4   $7,500.00
7   Honda   Blue          54738      4   $7,000.00
8  Toyota  White          60000      4   $6,250.00
9  Nissan  White          31600      4   $9,700.00 >
```

In [18]:
```
1 # functions: it has bracket
2 car_sales.describe()
```

Out[18]:

	Odometer (KM)	Doors
count	10.000000	10.000000
mean	78601.400000	4.000000
std	61983.471735	0.471405
min	11179.000000	3.000000
25%	35836.250000	4.000000
50%	57369.000000	4.000000
75%	96384.500000	4.000000
max	213095.000000	5.000000

In [19]:
```
1 car_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Make            10 non-null     object
 1   Colour          10 non-null     object
 2   Odometer (KM)   10 non-null     int64 
 3   Doors           10 non-null     int64 
 4   Price           10 non-null     object
dtypes: int64(2), object(3)
memory usage: 528.0+ bytes
```

In [20]:
```
1 car_sales.mean()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_5644\4073448239.py:1: FutureWarning: Dropping of nuisance col
umns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will rai
se TypeError.  Select only valid columns before calling the reduction.
  car_sales.mean()
```

Out[20]:
```
Odometer (KM)    78601.4
Doors                4.0
dtype: float64
```

In [21]:
```
1 car_prices = pd.Series([3000, 1500, 111250])
2 car_prices.mean()
```

Out[21]:
```
38583.333333333336
```

In [22]:
```
1 car_sales.sum()
```

Out[22]:
```
Make             ToyotaHondaToyotaBMWNissanToyotaHondaHondaToyo...
Colour               WhiteRedBlueBlackWhiteGreenBlueBlueWhiteWhite
Odometer (KM)                                               786014
Doors                                                           40
Price            $4,000.00 $5,000.00 $7,000.00 $22,000.00 $3,50...
dtype: object
```

In [23]:
```
1 car_sales['Doors'].sum()
```

Out[23]: 40

In [24]:
```
1 len(car_sales)
```

Out[24]: 10

In [25]:
```
1 car_sales
```

Out[25]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

viewing and selecting data

In [26]:
```
1 car_sales.head()
```

Out[26]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00

In [27]:
```
1 car_sales
```

Out[27]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [28]:
```
1 car_sales.head(7)
```

Out[28]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00

In [29]:
```
1 car_sales.tail()
```

Out[29]:

	Make	Colour	Odometer (KM)	Doors	Price
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [30]:
```
1 car_sales.tail(3)
```

Out[30]:

	Make	Colour	Odometer (KM)	Doors	Price
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [31]:
```
# .loc & .iloc
animals = pd.Series(['cat', 'dog', 'pandas', 'snake'])
animals
```

Out[31]:
```
0       cat
1       dog
2    pandas
3     snake
dtype: object
```

In [32]:
```
animals = pd.Series(['cat', 'dog', 'pandas', 'snake', 'birds'],
                    index = ([0, 3, 9, 8, 3]))
animals
```

Out[32]:
```
0       cat
3       dog
9    pandas
8     snake
3     birds
dtype: object
```

In [33]:
```
animals.loc[3]
```

Out[33]:
```
3      dog
3    birds
dtype: object
```

In [34]:
```
animals.loc[9]
```

Out[34]:
```
'pandas'
```

In [35]:
```
car_sales.loc[3]
```

Out[35]:
```
Make                    BMW
Colour                Black
Odometer (KM)         11179
Doors                     5
Price           $22,000.00
Name: 3, dtype: object
```

In [36]:
```
# .iloc refers to position
animals.iloc[3]
```

Out[36]:
```
'snake'
```

In [37]:
```
animals
```

Out[37]:
```
0       cat
3       dog
9    pandas
8     snake
3     birds
dtype: object
```

In [38]:
```
car_sales.iloc[3]
```

Out[38]:
```
Make                    BMW
Colour                Black
Odometer (KM)         11179
Doors                     5
Price           $22,000.00
Name: 3, dtype: object
```

In [39]:
```
car_sales
```

Out[39]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [40]:
```
animals
```

Out[40]:
```
0       cat
3       dog
9    pandas
8     snake
3     birds
dtype: object
```

In [41]:
```
animals.iloc[:3]
```

Out[41]:
```
0       cat
3       dog
9    pandas
dtype: object
```

In [42]:
```
1 car_sales.loc[:3]
```

Out[42]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00

In [43]:
```
1 car_sales.head(4)
```

Out[43]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00

In [44]:
```
1 car_sales['Make']
```

Out[44]:
```
0    Toyota
1     Honda
2    Toyota
3       BMW
4    Nissan
5    Toyota
6     Honda
7     Honda
8    Toyota
9    Nissan
Name: Make, dtype: object
```

In [45]:
```
1 car_sales['Colour']
```

Out[45]:
```
0    White
1      Red
2     Blue
3    Black
4    White
5    Green
6     Blue
7     Blue
8    White
9    White
Name: Colour, dtype: object
```

In [46]:
```
1 car_sales['Make']
```

Out[46]:
```
0    Toyota
1     Honda
2    Toyota
3       BMW
4    Nissan
5    Toyota
6     Honda
7     Honda
8    Toyota
9    Nissan
Name: Make, dtype: object
```

In [47]:
```
1 car_sales.Make
```

Out[47]:
```
0    Toyota
1     Honda
2    Toyota
3       BMW
4    Nissan
5    Toyota
6     Honda
7     Honda
8    Toyota
9    Nissan
Name: Make, dtype: object
```

In [48]:
```
1 car_sales['Odometer (KM)']
```

Out[48]:
```
0    150043
1     87899
2     32549
3     11179
4    213095
5     99213
6     45698
7     54738
8     60000
9     31600
Name: Odometer (KM), dtype: int64
```

In [49]:
```
1 car_sales
```

Out[49]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [50]:
```
car_sales[car_sales['Make']== 'Toyota']
```

Out[50]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
2	Toyota	Blue	32549	3	$7,000.00
5	Toyota	Green	99213	4	$4,500.00
8	Toyota	White	60000	4	$6,250.00

In [51]:
```
car_sales[car_sales['Odometer (KM)'] > 100000]
```

Out[51]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
4	Nissan	White	213095	4	$3,500.00

In [52]:
```
car_sales[car_sales['Odometer (KM)'] < 100000]
```

Out[52]:

	Make	Colour	Odometer (KM)	Doors	Price
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [53]:
```
# comparing two columns
pd.crosstab(car_sales['Make'], car_sales['Doors'])

```

Out[53]:

Doors	3	4	5
Make			
BMW	0	0	1
Honda	0	3	0
Nissan	0	2	0
Toyota	1	3	0

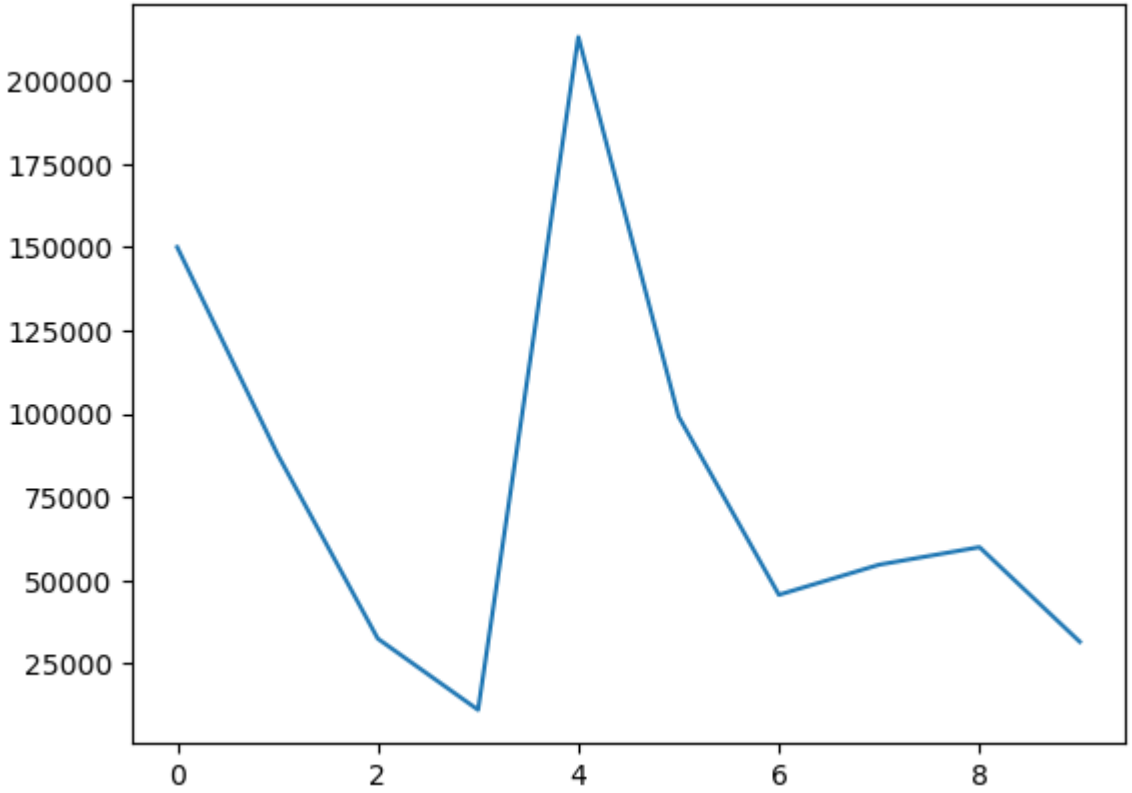
In [54]:
```
# comparing more columns: Groupby
car_sales.groupby(['Make']).mean()
```

Out[54]:

	Odometer (KM)	Doors
Make		
BMW	11179.000000	5.00
Honda	62778.333333	4.00
Nissan	122347.500000	4.00
Toyota	85451.250000	3.75

In [64]:
```
1 car_sales['Odometer (KM)'].plot()
```

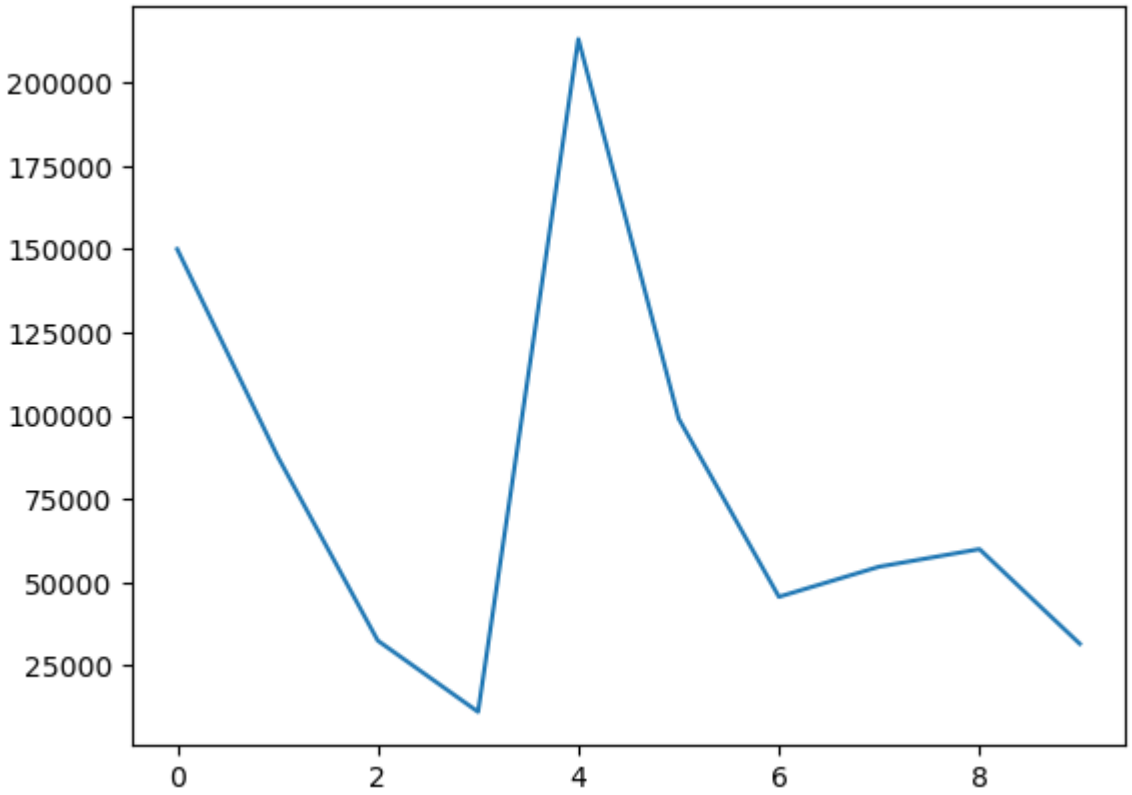
Out[64]: <AxesSubplot:>



In [65]:
```
1 # if the above code didn't run, we will use this to import matplotlib
2
3 %matplotlib inline
4 import matplotlib.pyplot as plt
```

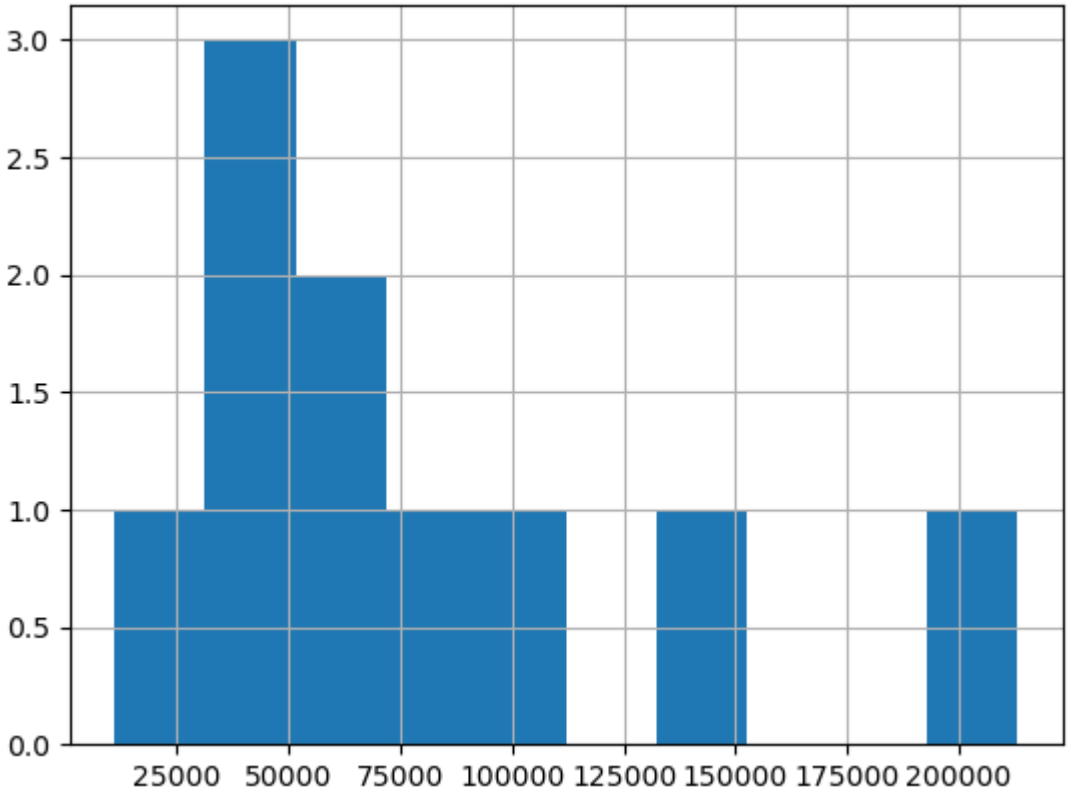
In [66]:
```
1 car_sales['Odometer (KM)'].plot()
```

Out[66]: <AxesSubplot:>



In [67]:
```
# histagram plot
car_sales['Odometer (KM)'].hist()
```

Out[67]: <AxesSubplot:>



In [68]:
```
car_sales['Price'].dtype
```

Out[68]: dtype('O')

In [69]:
```
1 car_sales['Price'].plot()
2 # price can not run cos is not a numerical data
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5644\2559747142.py in <module>
----> 1 car_sales['Price'].plot()
      2 # price can not run cos is not a numerical data

~\anaconda3\lib\site-packages\pandas\plotting\_core.py in __call__(self, *args, **kwargs)
    970                     data.columns = label_name
    971
--> 972         return plot_backend.plot(data, kind=kind, **kwargs)
    973
    974     __call__.__doc__ = __doc__

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\__init__.py in plot(data, kind, **kwargs)
     69             kwargs["ax"] = getattr(ax, "left_ax", ax)
     70     plot_obj = PLOT_CLASSES[kind](data, **kwargs)
---> 71     plot_obj.generate()
     72     plot_obj.draw()
     73     return plot_obj.result

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py in generate(self)
    325     def generate(self):
    326         self._args_adjust()
--> 327         self._compute_plot_data()
    328         self._setup_subplots()
    329         self._make_plot()

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py in _compute_plot_data(self)
    504         # no non-numeric frames or series allowed
    505         if is_empty:
--> 506             raise TypeError("no numeric data to plot")
    507
    508         self.data = numeric_data.apply(self._convert_to_ndarray)

TypeError: no numeric data to plot
```

In [70]:
```
1 car_sales
```

Out[70]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	$4,000.00
1	Honda	Red	87899	4	$5,000.00
2	Toyota	Blue	32549	3	$7,000.00
3	BMW	Black	11179	5	$22,000.00
4	Nissan	White	213095	4	$3,500.00
5	Toyota	Green	99213	4	$4,500.00
6	Honda	Blue	45698	4	$7,500.00
7	Honda	Blue	54738	4	$7,000.00
8	Toyota	White	60000	4	$6,250.00
9	Nissan	White	31600	4	$9,700.00

In [71]:
```
1 car_sales['Price'].fillna(0)
```

Out[71]:
```
0     $4,000.00
1     $5,000.00
2     $7,000.00
3    $22,000.00
4     $3,500.00
5     $4,500.00
6     $7,500.00
7     $7,000.00
8     $6,250.00
9     $9,700.00
Name: Price, dtype: object
```

Convert the price column to a numerical data

In [72]:
```
car_sales['Price'] = car_sales['Price'].str.replace('[^\d.]', '') # (clean price column: remove every
car_sales['Price'] = car_sales['Price'].astype(float) # convert the price column to float
car_sales['Price'] = car_sales['Price'].astype(int) # convert the price colum to integer
car_sales
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_5644\106475593.py:1: FutureWarning: The default value of rege
x will change from True to False in a future version.
  car_sales['Price'] = car_sales['Price'].str.replace('[^\d.]', '') # (clean price column: remove every
non-numeric character)
```

Out[72]:

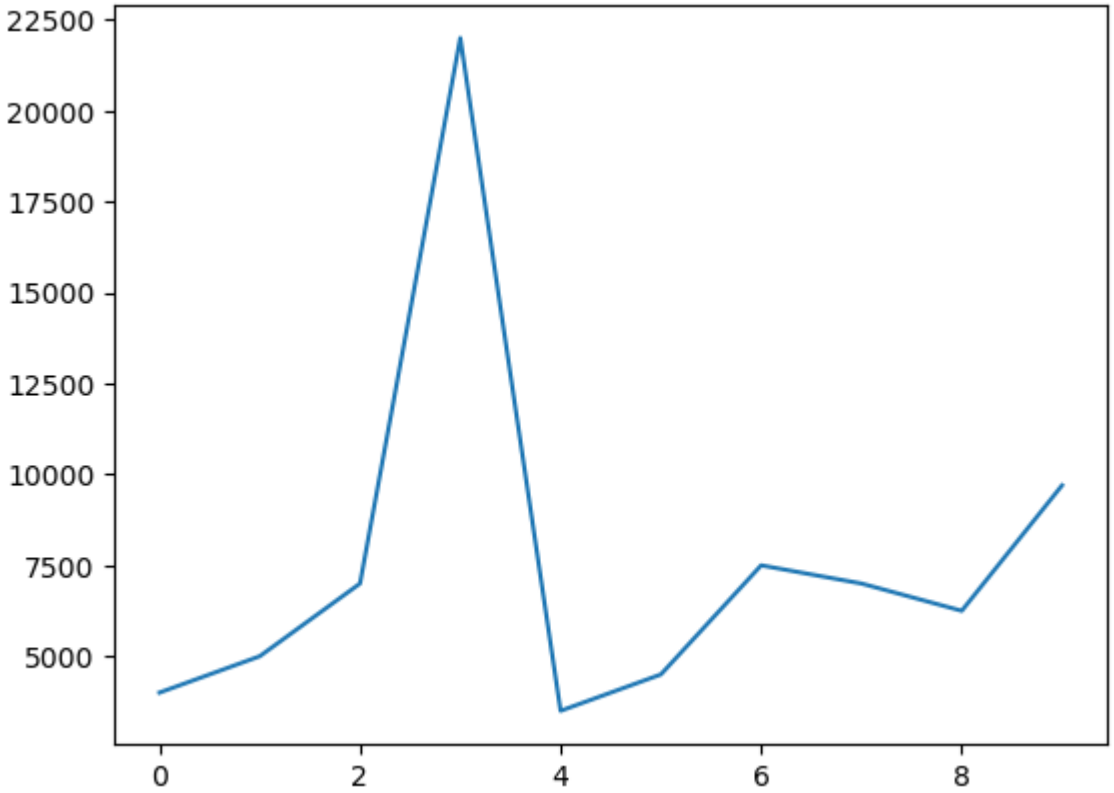
	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	150043	4	4000
1	Honda	Red	87899	4	5000
2	Toyota	Blue	32549	3	7000
3	BMW	Black	11179	5	22000
4	Nissan	White	213095	4	3500
5	Toyota	Green	99213	4	4500
6	Honda	Blue	45698	4	7500
7	Honda	Blue	54738	4	7000
8	Toyota	White	60000	4	6250
9	Nissan	White	31600	4	9700

In [73]:
```
1 car_sales['Price'].fillna(0)
```

Out[73]:
```
0     4000
1     5000
2     7000
3    22000
4     3500
5     4500
6     7500
7     7000
8     6250
9     9700
Name: Price, dtype: int32
```

In [74]:
```
car_sales['Price'].plot()
# price plot works perfectly
```

Out[74]: <AxesSubplot:>



Manipulating Data

In [75]:
```
1 car_sales['Make'].str.lower()
```

Out[75]:
```
0    toyota
1     honda
2    toyota
3       bmw
4    nissan
5    toyota
6     honda
7     honda
8    toyota
9    nissan
Name: Make, dtype: object
```

In [76]:
```
1 car_sales['Make'].str.upper()
```

Out[76]:
```
0    TOYOTA
1     HONDA
2    TOYOTA
3       BMW
4    NISSAN
5    TOYOTA
6     HONDA
7     HONDA
8    TOYOTA
9    NISSAN
Name: Make, dtype: object
```

In [79]:
```
1 car_sales['Make'] = car_sales['Make'].str.lower()
```

In [80]:
```
1 car_sales
```

Out[80]:

	Make	Colour	Odometer (KM)	Doors	Price
0	toyota	White	150043	4	4000
1	honda	Red	87899	4	5000
2	toyota	Blue	32549	3	7000
3	bmw	Black	11179	5	22000
4	nissan	White	213095	4	3500
5	toyota	Green	99213	4	4500
6	honda	Blue	45698	4	7500
7	honda	Blue	54738	4	7000
8	toyota	White	60000	4	6250
9	nissan	White	31600	4	9700

In [81]:
```
car_sales_missing = pd.read_csv('9.1 car-sales-missing-data.csv')
car_sales_missing
```

Out[81]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500
5	Toyota	Green	NaN	4.0	$4,500
6	Honda	NaN	NaN	4.0	$7,500
7	Honda	Blue	NaN	4.0	NaN
8	Toyota	White	60000.0	NaN	NaN
9	NaN	White	31600.0	4.0	$9,700

In [82]:
```
car_sales_missing['Odometer'].mean()
```

Out[82]:
```
92302.66666666667
```

In [83]:
```
# how to fill up the missing data
car_sales_missing['Odometer'].fillna(car_sales_missing['Odometer'].mean())
```

Out[83]:
```
0    150043.000000
1     87899.000000
2     92302.666667
3     11179.000000
4    213095.000000
5     92302.666667
6     92302.666667
7     92302.666667
8     60000.000000
9     31600.000000
Name: Odometer, dtype: float64
```

In [84]:
```
car_sales_missing # didn't work cos is not assigned
```

Out[84]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500
5	Toyota	Green	NaN	4.0	$4,500
6	Honda	NaN	NaN	4.0	$7,500
7	Honda	Blue	NaN	4.0	NaN
8	Toyota	White	60000.0	NaN	NaN
9	NaN	White	31600.0	4.0	$9,700

In [85]:
```
# it can be assigend with the following code:
# car_sales_missing['Odometer'] = car_sales_missing['Odometer'].fillna(car_sales_missing['Odometer'].
# car_sales_missing
```

In [86]:
```
car_sales_missing['Odometer'].fillna(car_sales_missing['Odometer'].mean(),
                                     inplace=True)
car_sales_missing
```

Out[86]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500
6	Honda	NaN	92302.666667	4.0	$7,500
7	Honda	Blue	92302.666667	4.0	NaN
8	Toyota	White	60000.000000	NaN	NaN
9	NaN	White	31600.000000	4.0	$9,700

In [87]:
```
car_sales_missing.dropna()
```

Out[87]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500

In [88]:
```
car_sales_missing
```

Out[88]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500
6	Honda	NaN	92302.666667	4.0	$7,500
7	Honda	Blue	92302.666667	4.0	NaN
8	Toyota	White	60000.000000	NaN	NaN
9	NaN	White	31600.000000	4.0	$9,700

In [89]:
```
car_sales_missing.dropna(inplace=True)
```

In [90]:
```
car_sales_missing
```

Out[90]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500

In [91]:
```
# to reaccess the original date, we reimport the data
car_sales_missing = pd.read_csv('9.1 car-sales-missing-data.csv')
car_sales_missing
```

Out[91]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500
5	Toyota	Green	NaN	4.0	$4,500
6	Honda	NaN	NaN	4.0	$7,500
7	Honda	Blue	NaN	4.0	NaN
8	Toyota	White	60000.0	NaN	NaN
9	NaN	White	31600.0	4.0	$9,700

In [92]:
```
# creating another data to drop the missing values
car_sales_missing_dropped = car_sales_missing.dropna()
car_sales_missing_dropped

```

Out[92]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500

In [93]:
```
car_sales_missing_dropped.to_csv('car-sales-missing-dropped.csv')
```

How do we create data from existing data

In [94]:
```
1 car_sales
```

Out[94]:

	Make	Colour	Odometer (KM)	Doors	Price
0	toyota	White	150043	4	4000
1	honda	Red	87899	4	5000
2	toyota	Blue	32549	3	7000
3	bmw	Black	11179	5	22000
4	nissan	White	213095	4	3500
5	toyota	Green	99213	4	4500
6	honda	Blue	45698	4	7500
7	honda	Blue	54738	4	7000
8	toyota	White	60000	4	6250
9	nissan	White	31600	4	9700

In [95]:
```
# column from series
seats_column = pd.Series([5, 5, 5, 5])

# new column called seats
car_sales['Seats'] = seats_column
car_sales
```

Out[95]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	toyota	White	150043	4	4000	5.0
1	honda	Red	87899	4	5000	5.0
2	toyota	Blue	32549	3	7000	5.0
3	bmw	Black	11179	5	22000	5.0
4	nissan	White	213095	4	3500	NaN
5	toyota	Green	99213	4	4500	NaN
6	honda	Blue	45698	4	7500	NaN
7	honda	Blue	54738	4	7000	NaN
8	toyota	White	60000	4	6250	NaN
9	nissan	White	31600	4	9700	NaN

In [96]:
```
1 car_sales
```

Out[96]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	toyota	White	150043	4	4000	5.0
1	honda	Red	87899	4	5000	5.0
2	toyota	Blue	32549	3	7000	5.0
3	bmw	Black	11179	5	22000	5.0
4	nissan	White	213095	4	3500	NaN
5	toyota	Green	99213	4	4500	NaN
6	honda	Blue	45698	4	7500	NaN
7	honda	Blue	54738	4	7000	NaN
8	toyota	White	60000	4	6250	NaN
9	nissan	White	31600	4	9700	NaN

In [97]:
```
car_sales['Seats'].fillna(5, inplace=True)
car_sales
```

Out[97]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	toyota	White	150043	4	4000	5.0
1	honda	Red	87899	4	5000	5.0
2	toyota	Blue	32549	3	7000	5.0
3	bmw	Black	11179	5	22000	5.0
4	nissan	White	213095	4	3500	5.0
5	toyota	Green	99213	4	4500	5.0
6	honda	Blue	45698	4	7500	5.0
7	honda	Blue	54738	4	7000	5.0
8	toyota	White	60000	4	6250	5.0
9	nissan	White	31600	4	9700	5.0

In [98]:
```
# column from python list
fuel_economy = ([7.5, 9.2, 5.0, 9.6, 8.7, 9.0, 9.0, 9.0, 9.0, 9.0])
car_sales['fuel per 100KM'] = fuel_economy
car_sales
```

Out[98]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM
0	toyota	White	150043	4	4000	5.0	7.5
1	honda	Red	87899	4	5000	5.0	9.2
2	toyota	Blue	32549	3	7000	5.0	5.0
3	bmw	Black	11179	5	22000	5.0	9.6
4	nissan	White	213095	4	3500	5.0	8.7
5	toyota	Green	99213	4	4500	5.0	9.0
6	honda	Blue	45698	4	7500	5.0	9.0
7	honda	Blue	54738	4	7000	5.0	9.0
8	toyota	White	60000	4	6250	5.0	9.0
9	nissan	White	31600	4	9700	5.0	9.0

In [99]:
```
car_sales['Total fuel used'] = car_sales['Odometer (KM)']/100 * car_sales['fuel per 100KM']
car_sales
```

Out[99]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Total fuel used
0	toyota	White	150043	4	4000	5.0	7.5	11253.225
1	honda	Red	87899	4	5000	5.0	9.2	8086.708
2	toyota	Blue	32549	3	7000	5.0	5.0	1627.450
3	bmw	Black	11179	5	22000	5.0	9.6	1073.184
4	nissan	White	213095	4	3500	5.0	8.7	18539.265
5	toyota	Green	99213	4	4500	5.0	9.0	8929.170
6	honda	Blue	45698	4	7500	5.0	9.0	4112.820
7	honda	Blue	54738	4	7000	5.0	9.0	4926.420
8	toyota	White	60000	4	6250	5.0	9.0	5400.000
9	nissan	White	31600	4	9700	5.0	9.0	2844.000

In [100]:
```
# create a column from a single value
car_sales['Number of wheels'] = 4
car_sales
```

Out[100]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Total fuel used	Number of wheels
0	toyota	White	150043	4	4000	5.0	7.5	11253.225	4
1	honda	Red	87899	4	5000	5.0	9.2	8086.708	4
2	toyota	Blue	32549	3	7000	5.0	5.0	1627.450	4
3	bmw	Black	11179	5	22000	5.0	9.6	1073.184	4
4	nissan	White	213095	4	3500	5.0	8.7	18539.265	4
5	toyota	Green	99213	4	4500	5.0	9.0	8929.170	4
6	honda	Blue	45698	4	7500	5.0	9.0	4112.820	4
7	honda	Blue	54738	4	7000	5.0	9.0	4926.420	4
8	toyota	White	60000	4	6250	5.0	9.0	5400.000	4
9	nissan	White	31600	4	9700	5.0	9.0	2844.000	4

In [101]:
```
car_sales['Passed road safety'] = True
car_sales
```

Out[101]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Total fuel used	Number of wheels	Passed road safety
0	toyota	White	150043	4	4000	5.0	7.5	11253.225	4	True
1	honda	Red	87899	4	5000	5.0	9.2	8086.708	4	True
2	toyota	Blue	32549	3	7000	5.0	5.0	1627.450	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	1073.184	4	True
4	nissan	White	213095	4	3500	5.0	8.7	18539.265	4	True
5	toyota	Green	99213	4	4500	5.0	9.0	8929.170	4	True
6	honda	Blue	45698	4	7500	5.0	9.0	4112.820	4	True
7	honda	Blue	54738	4	7000	5.0	9.0	4926.420	4	True
8	toyota	White	60000	4	6250	5.0	9.0	5400.000	4	True
9	nissan	White	31600	4	9700	5.0	9.0	2844.000	4	True

In [102]:
```
car_sales.dtypes
```

Out[102]:
```
Make                 object
Colour               object
Odometer (KM)         int64
Doors                 int64
Price                 int32
Seats               float64
fuel per 100KM      float64
Total fuel used     float64
Number of wheels      int64
Passed road safety     bool
dtype: object
```

In [103]:
```
# to drop a column
car_sales = car_sales.drop('Total fuel used', axis=1)
car_sales
```

Out[103]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
0	toyota	White	150043	4	4000	5.0	7.5	4	True
1	honda	Red	87899	4	5000	5.0	9.2	4	True
2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	4	True
4	nissan	White	213095	4	3500	5.0	8.7	4	True
5	toyota	Green	99213	4	4500	5.0	9.0	4	True
6	honda	Blue	45698	4	7500	5.0	9.0	4	True
7	honda	Blue	54738	4	7000	5.0	9.0	4	True
8	toyota	White	60000	4	6250	5.0	9.0	4	True
9	nissan	White	31600	4	9700	5.0	9.0	4	True

In [104]:
```
# right way to shuffle your dataframe

car_sales.sample(frac=1)
```

Out[104]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
8	toyota	White	60000	4	6250	5.0	9.0	4	True
1	honda	Red	87899	4	5000	5.0	9.2	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	4	True
9	nissan	White	31600	4	9700	5.0	9.0	4	True
5	toyota	Green	99213	4	4500	5.0	9.0	4	True
4	nissan	White	213095	4	3500	5.0	8.7	4	True
6	honda	Blue	45698	4	7500	5.0	9.0	4	True
0	toyota	White	150043	4	4000	5.0	7.5	4	True
2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
7	honda	Blue	54738	4	7000	5.0	9.0	4	True

In [105]:
```
# to re-asign
car_sales_shuffle = car_sales.sample(frac=1)
car_sales_shuffle
```

Out[105]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
6	honda	Blue	45698	4	7500	5.0	9.0	4	True
0	toyota	White	150043	4	4000	5.0	7.5	4	True
9	nissan	White	31600	4	9700	5.0	9.0	4	True
5	toyota	Green	99213	4	4500	5.0	9.0	4	True
8	toyota	White	60000	4	6250	5.0	9.0	4	True
2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
7	honda	Blue	54738	4	7000	5.0	9.0	4	True
4	nissan	White	213095	4	3500	5.0	8.7	4	True
1	honda	Red	87899	4	5000	5.0	9.2	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	4	True

In [106]:
```
# only select 20% of data
car_sales_shuffle.sample(frac=0.2)
```

Out[106]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	4	True

In [107]:
```
car_sales_shuffle.sample(frac=0.1)
```

Out[107]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
1	honda	Red	87899	4	5000	5.0	9.2	4	True

In [108]:
```
1 car_sales_shuffle.sample(frac=0.01)
```

Out[108]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety

In [109]:
```
1 car_sales_shuffle
```

Out[109]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
6	honda	Blue	45698	4	7500	5.0	9.0	4	True
0	toyota	White	150043	4	4000	5.0	7.5	4	True
9	nissan	White	31600	4	9700	5.0	9.0	4	True
5	toyota	Green	99213	4	4500	5.0	9.0	4	True
8	toyota	White	60000	4	6250	5.0	9.0	4	True
2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
7	honda	Blue	54738	4	7000	5.0	9.0	4	True
4	nissan	White	213095	4	3500	5.0	8.7	4	True
1	honda	Red	87899	4	5000	5.0	9.2	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	4	True

In [110]:
```
1 car_sales_shuffle.reset_index()
```

Out[110]:

	index	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
0	6	honda	Blue	45698	4	7500	5.0	9.0	4	True
1	0	toyota	White	150043	4	4000	5.0	7.5	4	True
2	9	nissan	White	31600	4	9700	5.0	9.0	4	True
3	5	toyota	Green	99213	4	4500	5.0	9.0	4	True
4	8	toyota	White	60000	4	6250	5.0	9.0	4	True
5	2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
6	7	honda	Blue	54738	4	7000	5.0	9.0	4	True
7	4	nissan	White	213095	4	3500	5.0	8.7	4	True
8	1	honda	Red	87899	4	5000	5.0	9.2	4	True
9	3	bmw	Black	11179	5	22000	5.0	9.6	4	True

In [111]:
```
car_sales_shuffle.reset_index(drop = True, inplace=True)
car_sales_shuffle
```

Out[111]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
0	honda	Blue	45698	4	7500	5.0	9.0	4	True
1	toyota	White	150043	4	4000	5.0	7.5	4	True
2	nissan	White	31600	4	9700	5.0	9.0	4	True
3	toyota	Green	99213	4	4500	5.0	9.0	4	True
4	toyota	White	60000	4	6250	5.0	9.0	4	True
5	toyota	Blue	32549	3	7000	5.0	5.0	4	True
6	honda	Blue	54738	4	7000	5.0	9.0	4	True
7	nissan	White	213095	4	3500	5.0	8.7	4	True
8	honda	Red	87899	4	5000	5.0	9.2	4	True
9	bmw	Black	11179	5	22000	5.0	9.6	4	True

In [112]:
```
# apply a function to a column
car_sales
```

Out[112]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
0	toyota	White	150043	4	4000	5.0	7.5	4	True
1	honda	Red	87899	4	5000	5.0	9.2	4	True
2	toyota	Blue	32549	3	7000	5.0	5.0	4	True
3	bmw	Black	11179	5	22000	5.0	9.6	4	True
4	nissan	White	213095	4	3500	5.0	8.7	4	True
5	toyota	Green	99213	4	4500	5.0	9.0	4	True
6	honda	Blue	45698	4	7500	5.0	9.0	4	True
7	honda	Blue	54738	4	7000	5.0	9.0	4	True
8	toyota	White	60000	4	6250	5.0	9.0	4	True
9	nissan	White	31600	4	9700	5.0	9.0	4	True

In [113]:
```
car_sales['Odometer (KM)'] =  car_sales['Odometer (KM)'].apply(lambda x: x / 1.6)
car_sales
```

Out[113]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	fuel per 100KM	Number of wheels	Passed road safety
0	toyota	White	93776.875	4	4000	5.0	7.5	4	True
1	honda	Red	54936.875	4	5000	5.0	9.2	4	True
2	toyota	Blue	20343.125	3	7000	5.0	5.0	4	True
3	bmw	Black	6986.875	5	22000	5.0	9.6	4	True
4	nissan	White	133184.375	4	3500	5.0	8.7	4	True
5	toyota	Green	62008.125	4	4500	5.0	9.0	4	True
6	honda	Blue	28561.250	4	7500	5.0	9.0	4	True
7	honda	Blue	34211.250	4	7000	5.0	9.0	4	True
8	toyota	White	37500.000	4	6250	5.0	9.0	4	True
9	nissan	White	19750.000	4	9700	5.0	9.0	4	True

In [ ]:
```
1
```